CIRCUIT FAMILIES

1) STATIC CMOS
2) RATIOED CIRCUITS
3) CASCODE VOLTAGE SWITCH LOGIC (CVSL)
4) DYNAMIC CIRCUITS
5) PASS TRANSISTOR CIRCUITS

STATIC CMOS

BUBBLE PUSHING (DEMORGAN'S LAW)
Compute \( F = AB + C \)

Using NANDS \& NORs

Start ANDs + ORs

\[ \begin{align*}
A & \quad D \\
B & \quad D \\
C &
\end{align*} \]

\[ \begin{align*}
A & \quad D \\
B & \quad D \\
C &
\end{align*} \]

\( \Rightarrow \)

Insert Double Inversions

\[ \begin{align*}
A & \quad D \\
B & \quad D \\
C &
\end{align*} \]

\[ \begin{align*}
A & \quad D \\
B & \quad D \\
C &
\end{align*} \]

\( \Rightarrow \)

Push Bubbles

Final
- Input ordering delay

\[ A \rightarrow 2 \rightarrow B \rightarrow 2 \rightarrow Y \]

Delay from "A" to "Y" less than "B" to "Y".

\[ A \text{ input cap = 4} \]

Asymmetric gates

\[ A \xrightarrow{\text{reset}} D \xrightarrow{D} Y \]

\[ \begin{align*}
Y &= A & \text{reset} &= 0 \\
Y &= D & \text{reset} &= 1
\end{align*} \]

Assuming "A" to "Y" delay critical.

\[ \overline{\text{reset}} \text{ to } "Y" \text{ not critical at all} \]

\[ \overline{\text{reset}} \]

\[ A \text{ input cap } = \frac{4}{3} + 2 = 3 \frac{1}{3} \]

\[ A \rightarrow \frac{4}{3} \rightarrow 4 \rightarrow Y \equiv \frac{1}{3} \]
SYMMETRIC NAND

Same delay for "A" and "B" inputs

- RATIO CIRCUITS (previously discussed)

- Static current draw
- Make pull-up "small" so small current draw when output low
CVSL (Cascode Voltage Switch Logic)

Generates both $Y + \overline{Y}$

No static power

(Ratioed Logic)

- Interesting approach but process sensitive makes slower than CMOS
- Slow NMOS, Fast PMOS $\Rightarrow$ Make PMOS small so NMOS can overcome it

Slow PMOS $\Rightarrow$ Slow pull-up

Generally slower than CMOS. More power hungry
Dynamic Circuits

- Requires clock \( \phi \) to operate
- Has precharge phase and evaluation phase
- Output only valid during eval phase
- Output precharged to \( V_{DD} \) during precharge phase
- Fast, no static power, high dynamic power

Inverter

- Footed allows input \( "A" \) to be \( "1" \) during precharge phase
- Unfooted requires \( "A" \) to be \( "0" \) during precharge phase
Strict requirement on inputs

- Inputs must be monotonic rising
  one of...
  - Low and stay low
  - Low and rise high
  - High and stay high \( \text{allowed if footed} \)
  - High and go low \( \text{not if unfooted} \)

But not high and go low

Since input is precharged high and cannot go high again if discharged

- Footed allows CMOS gates to drive dynamic gates
- Cannot connect Y directly to another gate as it violates above condition.

- Insert CMOS inverter between gates

$$\Rightarrow \text{Domino Logic (uses dynamic gates)}$$

\[ \Phi \quad \Phi \quad \Phi \quad \Phi \quad \Phi \]

\[ A \quad A \quad A \quad A \quad A \]

\[ B \quad B \quad B \quad B \quad B \]

\[ C \quad C \quad C \quad C \quad C \]

\[ \text{Domino AND} \quad \text{Domino AND} \]

\[ \text{OUT} \]
CAN BUILD MORE COMPLEX GATES

8 - INPUT MULTIPLEXER USING 2
4 - INPUT MUXES

ONE OF S0 -> S7 HIGH
**Domino Gates** inherently **non-inverting**

- *If last gate can use CMOS inverter*

- *If within domino gates, use dual-rail domino logic*

**All input/output signals encoded as a pair of signals**

<table>
<thead>
<tr>
<th>SIG H</th>
<th>SIG L</th>
<th>MEANING</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td><strong>Precharge</strong></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>&quot;0&quot;</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>&quot;1&quot;</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td><strong>Invalid</strong></td>
</tr>
</tbody>
</table>

**Similar to CVSL**

NAND

\[ Y_L - 0 \]
\[ A_L - 1 \]
\[ B_L - 1 \]

\[ \Phi - 1 \]
- Clock $\phi$ needs to always run for logic to operate.

- Dynamic power high due to dual-rail and activity factor high due to precharge phase.

- Keepers on eval phase.

- If $\phi$ stopped, charge leakage on dynamic node can change logic value.

- Use small $W_L$ PMOS transistor to overcome leakage.

![Circuit Diagram]

- $K$ small.
CHARGE-SHARING ERRORS

If node $x$ not precharged but $v_x = 0$ + node $v_y$ precharged to $V_{DD}$

Charge sharing occurs if $B = 0$

$A = 1$

$v_x = v_y = \frac{C_y}{C_y + C_x} V_{DD}$

Might drop $v_y$ below $v_{th}$ of $I_1$

And error occurs

- Might precharge all or some internal nodes to reduce this effect